Amendments to the Claims

- At the time of the action: Claims 21-41.
- Amended Claims: Claims 21-41.
- After this response: Claims 21-41.

1—20. (Previously Canceled).

21. (Currently Amended) A method of managing a write request from a first ~~source~~ compute node in a storage area network to a first storage node in the storage area network, comprising:

if there is an available direct communication ~~path~~ link between the first ~~source~~ compute node, a first switch, and the first storage node, then executing the write request from the first ~~source~~ compute node to the first storage node using the available direct communication link ~~path~~;

if there is not an available direct communication ~~path~~ link between the first ~~source~~ compute node and the first storage node, then:

transmitting the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available direct communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available direct communication ~~path~~ link from the second ~~source~~ compute node through the first switch or a second switch to the first storage node.

22. (Currently Amended) The method of claim 21, wherein if executing the write request from the first ~~source~~ compute node to the first storage node generates a timeout failure, then:

transmitting the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available communication path from the second ~~source~~ compute node to the first storage node.

23. (Currently Amended) The method of claim 22, wherein transmitting the write request from the first ~~source~~ compute node to the second ~~source~~ compute node comprises encapsulating the write request.

24. (Currently Amended) The method of claim 21, further comprising executing the write request from the second ~~source~~ compute node to the first storage node.

25. (Currently Amended) The method of claim 24, further comprising transmitting an error message from the second ~~source~~ compute node to the first ~~source~~ compute node if the write request fails.

26. (Currently Amended) A method of managing a write request from a first ~~source~~ compute node in a storage area network to a mirrored storage data set having a first storage node and a second storage node in the storage area network, comprising:

if there are available communication paths between the first ~~source~~ compute node and both the first storage node and the second storage node in the mirrored data set, then executing the write request from the first ~~source~~ compute node to both the first storage node and the second storage node using the available communication paths;

if there are no available communication paths between the first ~~source~~ compute node and the first storage node and the second storage node, then invoking an error routine;

if there is an available communication path between the first ~~source~~ compute node and only one of the first storage node and the second storage node in the mirrored data set, then:

executing the write request from the first ~~source~~ compute node to the first storage node or the second storage node via the available communication path;

transmitting the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available direct communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available communication path from the second ~~source~~ node through a first switch or a second switch to the first storage node or the second storage node.

27. (Currently Amended) The method of claim 26, wherein if executing the write request from the first ~~source~~ compute node to the first storage node generates a timeout failure, then:

transmitting the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available communication path from the second ~~source~~ compute node to the first storage node.

28. (Currently Amended) The method of claim 27, further comprising executing the write request from the second ~~source~~ compute node to the first storage node.

29. (Currently Amended) The method of claim 26, wherein if executing the write request from the first ~~source~~ compute node to the second storage node generates a timeout failure, then:

transmitting the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available communication path from the second ~~source~~ compute node to the second storage node.

30. (Currently Amended) The method of claim 29, further comprising executing the write request from the second ~~source~~ node to the first storage node.

31. (Currently Amended) A method of performing a surrogate write operation in a storage area network, comprising:

receiving, at a second ~~source~~ compute node, a query from a first ~~source~~ compute node, wherein the query identifies a target node in the storage network for the surrogate write operation;

transmitting a reply to the first ~~source~~ compute node, wherein the reply includes a signal component indicating there is an available communication path between the second ~~source~~ compute node and the target node; and

relaying write operations from the first ~~source~~ compute node to the target node.

32. (Currently Amended) The method of claim 31, further comprising determining whether there is an available communication path between the second ~~source~~ compute node and the target node.

33. (Currently Amended) The method of claim 31, wherein relaying write operations from the ~~source~~ compute node to the target node comprises:

receiving an encapsulated write request from the first ~~source~~ compute node;

de-encapsulating the encapsulated write request; and

executing the write request from the second node to the target node.

34. (Currently Amended) The method of claim 31, further comprising transmitting a failure signal from the second ~~source~~ compute node to the first ~~source~~ compute node if the write request from the second ~~source~~ compute node to the target node fails.

35. (Currently Amended) One or more computer-readable media comprising logic instructions for managing a write request from a first ~~source~~ compute node in a storage area network to a first storage node in the storage area network, that, when executed by a processor, cause the processor to perform operations comprising:

executing a write request from the first ~~source~~ compute node to the first storage node using an available communication path between the first ~~source~~ compute node and the first storage node;

if there is not an available communication path between the first ~~source~~ compute node and the first storage node, then:

transmitting the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available direct communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available direct communication path from the second ~~source~~ compute node through a first switch or a second switch to the first storage node.

36. (Currently Amended) The one or more computer-readable media of claim 35, further comprising logic instructions that, when executed by a processor, cause the processor to:

determine if executing the write request from the first ~~source~~ compute node to the first storage node generates a timeout failure, and if so, then to transmit the write request from the first ~~source~~ compute node to a second ~~source~~ compute node if there is an available communication path from the first ~~source~~ compute node to the second ~~source~~ compute node and an available communication path from the second ~~source~~ compute node to the first storage node.

37. (Currently Amended) The one or more computer-readable media of claim 36, further comprising logic instructions that, when executed by a processor, cause the processor to encapsulate the write request before transmitting the write request from the first ~~source~~ compute node to the second ~~source~~ compute node.

38. (Currently Amended) One or more computer-readable media comprising logic instructions for performing a surrogate write operation in a storage area network that, when executed by a processor, cause the processor to perform operations comprising:

receiving, at a second ~~source~~ compute node, a query from a first ~~source~~ compute node, wherein the query identifies a target node in the storage area network for the surrogate write operation;

transmitting a reply to the first ~~source~~ compute node, wherein the reply includes a signal component indicating there is an available communication path between the second ~~source~~ compute node and the target node; and

relaying write operations from the first ~~source~~ compute node to the target node.

39. (Currently Amended) The one or more computer-readable media of claim 38, further comprising logic instructions that, when executed on a processor, cause the processor to determine whether there is an available communication path between the second ~~source~~ node and the target node.

40. (Currently Amended) The one or more computer-readable media of claim 38, further comprising logic instructions that, when executed on a processor, cause the processor to perform operations comprising:

receiving an encapsulated write request from the first ~~source~~ compute node;

de-encapsulating the encapsulated write request; and

executing the write request from the second node to the target node.

41. (Currently Amended) The one or more computer-readable media of claim 38, further comprising logic instructions that, when executed on a processor, cause the processor to transmit a failure signal from the second ~~source~~ compute node to the first ~~source~~ compute node if the write request from the second ~~source~~ compute node to the target node fails.